CLAIMS

WHAT IS CLAIMED IS:

1. A discrete event simulator comprising:

an event scheduler for sorting incoming future events; and

at least one simulation engine for evaluating pending events, said at least one simulation engine being remote from said scheduler but being communicatively connected thereto.

2. The discrete event simulator of Claim 1, wherein said at least one simulation engine is capable evaluating pending events which include a model core selected from the group consisting of a truth table, a lookup, a reduction to finite conclusion of a continuous or transient equation, a re-programmable core-processor, a software emulated core within the model, and a hardware emulated core.

3. The discrete event simulator of Claim 1, wherein said at least one simulation engine comprises a logic simulation engine.

4. The discrete event simulator of Claim 1, wherein said at least one simulation engine comprises a memory simulation engine.

5. The discrete event simulator of Claim 1, wherein said at least one simulation engine comprises a software simulation engine.

6. The discrete event simulator of Claim 1, wherein said at least one simulation engine comprises an interconnections simulation engine.

7. The discrete event simulator of Claim 1, wherein said at least one simulation engine comprises more than one simulation engines, further wherein each of said more than one simulation engines is directly connected to the event scheduler.

8. The discrete event simulator of Claim 7, wherein each of said more than one simulation engines is selected from the group consisting of a logic engine, a memory engine, a software engine, an interconnection engine, and subsets of the above engines.

9. The discrete event simulator of Claim 7, wherein at least two of said more than one simulation engines are of the same type.

10. The discrete event simulator of Claim 1, further comprising:

at least one pending event queue directly connecting the scheduler to said at least one simulation engine; and

at least one future event queue directly connecting said at least one simulation engine to said scheduler.

11. The discrete event simulator of Claim 10, wherein said pending event and said future event queues are conventional computer bus structures.

12. The discrete event simulator of Claim 10, wherein said pending event and said future event queues are point-to-point connections between the scheduler and each of said at least one simulation engines.

13. The discrete event simulator of Claim 12, wherein said point-to-point connections are gigabit conduits.

14. The discrete event simulator of Claim 1, wherein said scheduler comprises at least one timing wheel.

15. The discrete event simulator of Claim 1, wherein said scheduler comprises at least two timing wheels.

16. The discrete event simulator of Claim 1, wherein said scheduler further comprises:

an event RAM for holding a queue of received events; and

a pointer RAM for holding head of queue pointers which point to the first element in a queue of events for each simulation time period.

17. The discrete event simulator of Claim 16, wherein said event RAM and said pointer RAM are dual ported.

18. The discrete event simulator of Claim 16, wherein said event RAM includes a data RAM and a next event RAM.

19. The discrete event simulator of Claim 16, wherein said event RAM is wide enough to read/write an entire event in one clock cycle.

20. The discrete event simulator of Claim 19, wherein said event RAM is 144 bits wide.

21. The discrete event simulator of Claim 16, wherein said event RAM includes rollover functionality to reuse memory locations that have already had their data released as pending events to the simulation engine.

22. The discrete event simulator of Claim 16, wherein said event scheduler utilizes a global virtual time to keep track of simulation time and release all pending events at one time.

23. The discrete event simulator of Claim 1, wherein at least one of said at least one simulation engines comprises a pipelined structure such that an incoming event can be evaluated each clock cycle.

24. The discrete event simulator of Claim 1, wherein said at least one simulation engine is adapted to simulate the intrinsic delay within each structure being simulated.

25. The discrete event simulator of Claim 1, wherein said at least one simulation engine is adapted to simulate the extrinsic delay due to output load for each structure being simulated.

26. The discrete event simulator of Claim 1, wherein said at least one simulation engine is adapted to simulate the wire delay between each structure being simulated.

27. The discrete event simulator of Claim 1, further comprising:

a net-list engine communicatively coupled to said event scheduler, wherein said net-list engine is capable of generating additional events that

account for the capacitive effects caused by having multiple discrete event elements connected to a single output of a first discrete event element.

28. The discrete event simulator of Claim 27, wherein said net-list engine is located between an output queue of said simulation engine and said event scheduler.

29. The discrete event simulator of Claim 27, wherein said net-list engine is located between said event scheduler and an input queue of said simulation engine.

30. The discrete event simulator of Claim 27, wherein said net-list engine is directly connected to said event scheduler and is in parallel with said at least one simulation engine.

31. The discrete event simulator of Claim 1, further comprising:

a host workstation communicatively coupled to said event scheduler and said at least one simulation engine.

32. The discrete event simulator of Claim 31, wherein said host workstation is capable of altering the setup of the discrete system being simulated while the simulation is running.

33. The discrete event simulator of Claim 31, wherein said host computer is adapted to receive future events to analyze the system being simulated for runtime errors.

34. The discrete event simulator of Claim 4, wherein said memory simulation engine comprises:

a data storage RAM/CAM;

emulation logic; and

a configuration RAM.

35. The discrete event simulator of Claim 1, wherein elements of a discrete event system to be simulated are represented using a universal device primitive.

36. A discrete logic simulator, comprising:

an event scheduler for sorting incoming future events;

at least one simulation engine for evaluating pending events, said at least one simulation engine being remote from said scheduler but being communicatively connected thereto;

at least one pending event queue directly connecting the scheduler to said at least one simulation engine;

at least one future event queue directly connecting said at least one simulation engine to said scheduler; and

a host workstation communicatively coupled to said event scheduler and said at least one simulation engine.

37. The discrete logic simulator of Claim 36, wherein said at least one simulation engine is capable evaluating pending events which include a model core selected from the group consisting of a truth table, a lookup, a reduction to finite conclusion of a continuous or transient equation, a re-programmable core-processor, a software emulated core within the model, and a hardware emulated core.

38. An event scheduler for use with a discrete event simulator, comprising:

an event RAM for holding a queue of received events; and

a pointer RAM for holding head of queue pointers which point to the first element in a queue of events for each simulation time period.

39. The event scheduler of Claim 38, wherein said event RAM is partitioned into a data RAM and a next RAM.

40. A method for simulating a system of discrete events, comprising the steps of:

loading a discrete event system to be simulated into at least one simulation engine for evaluating pending events and creating future events to be evaluated at a later time;

loading initial conditions into said at least one simulation engine and an event scheduler remote from said simulation engine capable of sorting said future events;

starting said simulation; and

receiving results of said simulation.

41. The method of Claim 40, wherein each of these steps is facilitated by a host computer communicatively connected to said event scheduler and said at least one simulation engine.

42. The method of Claim 40, wherein said at least one simulation engine comprises more than one simulation engines, further wherein each of said more than one simulation engines is directly connected to the event scheduler.

43. The method of Claim 42, wherein each of said more than one simulation engines is selected from the group consisting of a logic engine, a memory engine, a software engine, an interconnection engine, and subsets of the above engines.

44. The method of Claim 40, wherein said scheduler further comprises:

an event RAM for holding a queue of received events; and

a pointer RAM for holding head of queue pointers which point to the first element in a queue of events for each simulation time period.

45. The method of Claim 40, wherein said first loading step includes storing representations of elements of the system of discrete events using universal device primitives.

46. A method for sorting and scheduling events in a discrete event simulator, comprising the steps of:

receiving a future event from one of at least one remote simulation engines, said event including a time stamp defining at what time in the future the event is to be evaluated by one of said remote simulation engines;

inserting said future event into an empty memory location indicated by a head of empty queue pointer; and

updating a head of queue pointer for a time period corresponding to the time stamp of the future event to add the future event into a queue of events to be evaluated at said time period.

47. The method of Claim 46, wherein simulation time is controlled using a global virtual time.

48. The method of Claim 47, further comprising the steps of:

incrementing said global virtual time; and

releasing a queue of saved events as pending events to one of said remote simulation engines, wherein said queue is indicated by a head of queue pointer for the time period matching the incremented global virtual time.

49. The method of Claim 48, further comprising the step of:

generating a plurality of gate events based on at least one of said sent pending events to account for capacitive effects of element fanout in said discrete event system.

50. The method of Claim 46, further comprising the step of:

generating a plurality of future events responsive to the capacitive effects of element fanout in said discrete event system, wherein said plurality of future events are based on a single event from one of said remote simulation engines.